AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently Amended) A computer-implemented method for processing software code, said method comprising:

loading a virtual machine engine into a common memory, the common memory accessible by a first processor and a second processor, wherein the first processor and the second processor are heterogeneous processors, wherein the first processor executes a first instruction set and wherein the second processor executes a second instruction set, and wherein the first processor includes a first local memory accessible to the first processor and the second processor includes a second local memory accessible to the second processor;

loading, by the first processor, virtual machine code into the common memory, the virtual machine code adapted to be processed by the virtual machine engine;

writing, by the first processor, a code processing request into a mailbox associated with the second processor;

receiving, by the second processor, the code processing request from the second processor's mailbox;

in response to receiving the code processing request, loading, at [[a]] the second processor, [[a]] the virtual machine ~~program~~ engine from the common memory into [[a]] the second local memory ~~corresponding to the second processor~~;

~~receiving, at the second processor, a code processing request requested by a first processor, wherein the first and second processors are heterogeneous processors within a computer system that share a common memory;~~

in further response to receiving the code processing request, reading, by the second processor from the common memory ~~shared by the first and second~~

~~processors,~~ software code data corresponding to the <u>code</u> request, the software code data including <u>the</u> virtual machine code ~~adapted to be processed by the virtual machine program~~;

writing the software code data corresponding to the <u>code processing</u> request to the <u>second</u> local memory ~~corresponding to the second processor in response to the request~~;

processing the software code data by the second processor, wherein the processing includes processing the virtual machine code at the second processor using the virtual machine ~~program~~ <u>engine</u>, the processing resulting in executable instructions<u>, the executable instructions comprising instructions from the first instruction set and capable of being executed by the first processor</u>;

writing the executable instructions to a memory location accessible by the first processor; and

executing, at the first processor, the executable instructions.

2.   (Canceled)

3.   (Original) The method as described in claim 1 further comprising:

writing data resulting from the executing to the common memory.

4.   (Currently Amended) The method as described in claim 1 further comprising:

prior to ~~the receiving~~ <u>loading the virtual machine code into the common memory</u>:

running a first <u>application</u> program;

in response to running the first <u>application</u> program, identifying a call to a software effect corresponding to the software code data; and

loading the software code data into the common memory, wherein the processing of the software code data occurs during the running of the first

application program and wherein the processing is completed prior to the first program calling the software effect.

5.   (Canceled)

6.   (Previously Presented) The method as described in claim 4 further comprising:

receiving, at the first processor, the executable instructions resulting from the processing performed by the second processor, wherein the executable instructions are adapted to perform a multimedia effect; and

performing the multimedia effect on the first processor by executing the received executable instructions.

7.   (Previously Presented) The method as described in claim 1 wherein the writing further comprises:

writing the executable instructions to a memory location accessible by the first processor using a direct memory access (DMA) operation.

8.   (Currently Amended) The method as described in claim 7 wherein the memory location is selected from the group consisting of [[a]] the first local memory corresponding to the first processor, and the common memory.

9.   (Canceled)

10.   (Currently Amended) The method as described in claim 1 wherein the processing results in one or more program instructions adapted to be performed by the first processor, the method further comprising:

writing the program executable instructions to the common memory; and

notifying the first processor that the program executable instructions have been written; and

~~executing the program instructions by the first processor.~~

11.   (Canceled)

12.   (Canceled)

13.   (Canceled)

14.   (Canceled)

15.   (Canceled)

16.   (Canceled)

17.   (Canceled)

18.   (Canceled)

19.   (Canceled)

20.   (Canceled)

21.   (Canceled)

22.   (Canceled)

23.   (Canceled)

24.   (Canceled)

25.   (Canceled)

26. (Canceled)

27. (Canceled)

28. (Canceled)

29. (Canceled)

30. (Canceled)